Clowdr Documentation

Greg Kiar, Tristan Glatard

Aug 09, 2019

Contents:

CHAPTER 1

Launching Local & Cluster Tasks

Manages local and cluster deployment. Ideal for development, testing, executing on local resources, or deployment on a computing cluster environment.

```
usage: clowdr local [-h] [--verbose] [--dev] [--workdir WORKDIR]
                    [--volumes VOLUMES] [--groupby GROUPBY] [--sweep SWEEP]
                    [--setup] [--cluster {slurm}] [--clusterargs CLUSTERARGS]
                    [--jobname JOBNAME] [--simg SIMG] [--user]
                    [--rerun {all,select,failed,incomplete}] [--run_id RUN_ID]
                    [--task_ids TASK_IDS [TASK_IDS ...]] [--s3 S3] [--bids]
                    descriptor invocation provdir
```

1.1 Positional Arguments

descriptor — Local path to Boutiques descriptor for the tool you wish to run. To learn about descriptors and Boutiques, go to: https://boutiques.github.io.

invocation — Local path to Boutiques invocation (or directory containing multiple invocations) for the analysis you wish to run. To learn about invocations and Boutiques, go to: https://boutiques.github.io.

provdir — Local directory for Clowdr provenance records and other captured metadata to be stored. This directory needs to exist prior to running Clowdr.

1.2 Named Arguments

--verbose, -V — Toggles verbose output statements.

Default: False

--dev, -d — Launches only the first created task. This is intended for development purposes.

Default: False

--workdir, -w Specifies the working directory to be used by the tasks created.

--volumes, -v Specifies any volumes to be mounted to the container. This is usually related to the path of any data files as specified in your invocation(s).

--groupby, -g If you wish to run tasks in batches, specify the number of tasks to group here. For imperfect multiples, the last group will be the remainder.

--sweep If you wish to perform a parameter sweep with Clowdr, you can use this flag and provide Boutiques parameter ID as the argument here. This requires: 1) the parameter exists in the provided invocation, and 2) that field contains a list of the parameter values to be used (if it is ordinarily a list, this means it must be a list of lists here). This option does not work with directories of invocations, but only single files.

--setup If you wish to generate metadata but not launch tasks then you can use this mode.

Default: False

--cluster, -c Possible choices: slurm

If you wish to submit your local tasks to a scheduler, you must specify it here. Currently this only supports SLURM clusters.

--clusterargs, -a This allows users to supply arguments to the cluster, such as specifying RAM or requesting a certain amount of time on CPU. These are provided in the form of key:value pairs, and separated by commas. For example: –clusterargs time:4:00,mem:2048,account:ABC

--jobname, -n If running on a cluster, and you wish to specify a unique identifier to appear in thesubmitted tasks, you can specify it with this flag.

--simg, -s If the Boutiques descriptor summarizes a tool wrapped in Singularity, and the image has already been downloaded, this option allows you to specify that image file.

--user, -u If the Boutiques descriptor summarizes a tool wrapped in Docker, toggles propagating the current user within the container.

Default: False

--rerun, -R Possible choices: all, select, failed, incomplete

Allows user to re-run jobs in a previous execution that either failed or didn't finish, etc. This requires the –run_id argument to also be supplied. Four choices are: 'all' to re-run all tasks, 'select' to re-run specific tasks, 'failed' to re-run tasks which finished with a non-zero exit-code, 'incomplete' to re-run tasks which have not yet indicated job completion. While the descriptor and invocations will be adopted from the previous executions, other options such as clusterargs or volume can be set to different values, if they were the source of errors. Pairing the incomplete mode with the –dev flag allows you to walk through your dataset one group at a time.

--run_id Pairs with –rerun. This ID is the directory within the supplied provdir which contains execution you wish to relaunch. These IDs/directories are in the form: year-month-day_hour-minute-second-8digitID.

--task_ids Pairs with –rerun. This list of task IDs are the task numbers within the directory supplied with –run_id and provdir. These IDs are integers greater than or equal to 0.

--s3 Amazon S3 bucket and path for remote data. Accepted in the format: s3://{bucket}/{path}

--bids, -b Indicates that the tool being launched is a BIDS app. BIDS is a data organization format in neuroimaging. For more information about this, go to https://bids.neuroimaging.io.

Default: False

CHAPTER 2

Launching Cloud Tasks

Manages cloud deployment. Ideal for running jobs at scale on data stored in Amazon Web Services S3 buckets (or similar object store).

```
usage: clowdr cloud [-h] [--verbose] [--dev] [--region REGION] [--sweep SWEEP]
                    [--bids]
                    descriptor invocation provdir s3 {aws} credentials
```

2.1 Positional Arguments

descriptor Local path to Boutiques descriptor for the tool you wish to run. To learn about descriptors and Boutiques, go to: https://boutiques.github.io.

invocation Local path to Boutiques invocation (or directory containing multiple invocations) for the analysis you wish to run. To learn about invocations and Boutiques, go to: https://boutiques.github.io.

provdir Local directory for Clowdr provenance records and other captured metadata to be stored. This directory needs to exist prior to running Clowdr.

s3 Amazon S3 bucket and path for remote data. Accepted in the format: s3://{bucket}/{path}

cloud Possible choices: aws

Specifies which cloud endpoint you'd like to use. Currently, only AWS is supported.

credentials Your credentials file for the resource.

2.2 Named Arguments

--verbose, -V Toggles verbose output statements.

	Default: False
--dev, -d	Launches only the first created task. This is intended for development purposes. Default: False
--region, -r	The Amazon region to use for processing.
--sweep	If you wish to perform a parameter sweep with Clowdr, you can use this flag and provide Boutiques parameter ID as the argument here. This requires: 1) the parameter exists in the provided invocation, and 2) that field contains a list of the parameter values to be used (if it is ordinarily a list, this means it must be a list of lists here). This option does not work with directories of invocations, but only single files.
--bids, -b	Indicates that the tool being launched is a BIDS app. BIDS is a data organization format in neuroimaging. For more information about this, go to https://bids.neuroimaging.io. Default: False

CHAPTER 3

Sharing Your Analysis

```
usage: clowdr share [-h] [--prepare] [--host HOST] [--port PORT] [--debug]
                    [--verbose]
                    provdir
```

3.1 Positional Arguments

provdir Local or S3 directory where Clowdr provenancerecords and metadata are stored. This path was returned by running either clowdr cloud or clowdr local. This can also be a clowdr-generated summary file.

3.2 Named Arguments

--prepare, -p If provided, this prevents a server from being launched after metadata is consolidated into a single file, and the path to that file is returned.

Default: False

--host The host to broadcast the share service at. Default is 0.0.0.0.

Default: "0.0.0.0"

--port The port to broadcast the share service at. Default is 8050.

Default: 8050

--debug, -d Toggles server messages and logging. This is intended for development purposes.

Default: False

--verbose, -V Toggles verbose output statements.

Default: False

CHAPTER 4

Manually Running Tasks

```
usage: clowdr task [-h] [--verbose] [--provdir PROVDIR] [--local]
                   [--workdir WORKDIR] [--volumes VOLUMES]
                   [--imagepath IMAGEPATH]
                   tasklist [tasklist ...]
```

4.1 Positional Arguments

tasklist One or more Clowdr-created task.json files summarizing the jobs to be run. These task files are created by one of clowdr cloud or clowdr local.

4.2 Named Arguments

--verbose, -V Toggles verbose output statements.

Default: False

--provdir, -p Local or directory where Clowdr provenance records and metadata will be stored. This is optional here because it will be stored by default in a temporary location and moved, unless this is specified.

--local, -l Flag indicator to identify whether the task is being launched on a cloud or local resource. This is important to ensure data is transferred off clouds before shut down.

Default: False

--workdir, -w Specifies the working directory to be used by the tasks created.

--volumes, -v Specifies any volumes to be mounted to the container. This is usually related to the path of any data files as specified in your invocation(s).

--imagepath	If the Boutiques descriptor summarizes a tool wrapped in Singularity, and the image has already been downloaded, this option allows you to specify that image file.

CHAPTER 5

Clowdr Python Interface

5.1 clowdr package

5.1.1 Subpackages

clowdr.controller package

Submodules

clowdr.controller.launcher module

clowdr.controller.launcher.**configureResource**(*endpoint*, *auth*, ***kwargs*)

clowdr.controller.metadata module

clowdr.controller.metadata.**bidsTasks**(*clowdrloc*, *taskdict*)

bidsTask Scans through BIDS app fields for creating more tasks than specified.

clowdrloc [str] Path for storing Clowdr intermediate files and outputs

taskdict [str] Dictionary of the tasks (pre-BIDS-ification)

tuple: (list, list) The task dictionary JSONs, and associated Boutiques invocation files.

clowdr.controller.metadata.**consolidateTask**(*tool*, *invocation*, *clowdrloc*, *dataloc*, *bids=False*, *sweep=[]*, *verbose=False*, ***kwargs*)

Creates Clowdr task JSON files and Boutiques invocations which summarize all associated metadata with the tasks being launched.

tool [str] Path to a boutiques descriptor for the tool to be run.

invocation [str] Path to a boutiques invocation for the tool and parameters to be run.

clowdrloc [str] Path for storing Clowdr intermediate files and output logs.

dataloc [str] Path for accessing input data on an S3 bucket (must include s3://) or localhost for non-cloud hosted data.

bids [bool (default = False)] Flag toggling BIDS-aware metadata preparation.

sweep [list (default = [])] List of parameters to sweep over in the provided invocations.

verbose [bool (default = False)] Flag toggling verbose output printing.

****kwargs** [dict] Arbitrary additional keyword arguments which may be passed.

tuple: (list, list) The task dictionary JSONs, and associated Boutiques invocation files.

clowdr.controller.metadata.**prepareForRemote**(*tasks*, *tmploc*, *clowdrloc*)
Scans through BIDS app fields for creating more tasks than specified.

tasks [list] List of task dictionaries on disk for Clowdr tasks.

tmploc [str] Temporary location where the invocations and task files are stored.

clowdrloc [str] Path for storing Clowdr intermediate files and outputs

tuple: (list, list) The task dictionary JSONs, and associated Boutiques invocation files, with paths corrected to eventual remote locations.

clowdr.controller.metadata.**sweepTasks**(*taskdicts*, *invocations*, *sweep_param*)
Sweeps through provided fields for creating more tasks than specified.

taskdicts [str] Dictionary of the tasks

invocations [str] Corresponding invocations for each task dictionary

sweep_param [str] Parameter to be swept over in each invocation

tuple: (list, list) The task dictionary JSONs, and associated Boutiques invocation files.

Module contents

clowdr.endpoint package

Submodules

clowdr.endpoint.AWS module

class clowdr.endpoint.AWS.**AWS**(*auth*)
Bases: *clowdr.endpoint.remote.Endpoint*

configureBatch(***kwargs*)

configureIAM(***kwargs*)

launchJob(*taskloc*)

setCredentials(***kwargs*)

startSession()

clowdr.endpoint.remote module

class clowdr.endpoint.remote.**Endpoint**(*auth*)
Bases: object

Module contents

5.1.2 Submodules

5.1.3 clowdr.driver module

clowdr.driver.**cloud**(*descriptor*, *invocation*, *provdir*, *s3*, *cloud*, *credentials*, ***kwargs*)
Launches a pipeline locally at scale through Clowdr.

descriptor [str] Path to a boutiques descriptor for the tool to be run

invocation [str] Path to a boutiques invocation for the tool and parameters to be run

provdir [str] Path on S3 for storing Clowdr intermediate files and outputs

s3 [str] Path on S3 for accessing input data

cloud [str] Which endpoint to use for deployment

credentials [str] Credentials for Amazon with access to dataloc, clowdrloc, and Batch

****kwargs** [dict] Arbitrary keyword arguments (i.e. {'verbose': True})

int The exit-code returned by the task being executed

clowdr.driver.**local**(*descriptor*, *invocation*, *provdir*, *backoff_time=36000*, *sweep=[]*, *verbose=False*, *workdir=None*, *simg=None*, *rerun=None*, *run_id=None*, *task_ids=[]*, *volumes=[]*, *s3=None*, *cluster=None*, *jobname=None*, *clusterargs=None*, *dev=False*, *groupby=1*, *user=False*, *setup=False*, *bids=False*, ***kwargs*)
cluster Launches a pipeline locally through the Clowdr wrappers.

descriptor [str] Path to a boutiques descriptor for the tool to be run.

invocation [str] Path to a boutiques invocation for the tool and parameters to be run.

provdir [str] Path for storing Clowdr intermediate files and output logs.

backoff_time [int (default = 36000)] Maximum delay time before attempting resubmission of jobs that failed to be submitted to a scheduler, in seconds.

sweep [list (default = [])] List of parameters to sweep over in the provided invocations.

verbose [bool (default = False)] Flag toggling verbose output printing

workdir [str (default = None)] Working directory to be used in execution, if different from provdir.

simg [str (default = None)] Path to local copy of Singularity image to be used during execution.

rerun [str (default = None)] One of "all", "select", "failed", and "incomplete," which enables re-launching tasks from a previous execution either individually or in commonly-desired groups.

run_id [str (default = None)] Required when using rerun, above, this specifies the experiment ID to be re-run. This is the directory created for metadata, of the form:

year-month-day_hour-minute-second-8digitID.

task_ids [list (default = [])] If re-running with the "select" mode, a list of task IDs within the directory specified by run_id which are to be re-run.

volumes [list (default = [])]

List of volume mount-path strings, specified using the standard: /path/on/host/:/path/in/container/

s3 [str (default = None)] Path for accessing input data on an S3 bucket. Must include s3://.

cluster [str (default = None)] Scheduler on the cluster being used. Currently only slurm is supported.

jobname [str (default = None)] Base-name for the jobs as they will appear in the scheduler.

clusterargs [str (default = None)] Comma-separated list of arguments to be provided to the cluster on job submission. Such as: time:4:00,mem:2048,account:ABC

dev [bool (default = False)] Flag to toggle dev mode which only runs the first execution in the set.

groupby [int (default = 1)] Value which dictates the grouping of tasks. Particularly useful when tasks are short or a cluster restricts the number of unique jobs.

user [bool (default = False)] When running with Docker, toggles whether or not the host-user's UID is used within the container.

setup [bool (default = False)] Flag which prevents execution of tasks after the metadata task and invocation files are generated.

bids [bool (default = False)] Flag toggling BIDS-aware metadata preparation.

****kwargs** [dict] Arbitrary additional keyword arguments which may be passed.

str The path to the created directory containing Clowdr experiment metadata.

clowdr.driver.**main**(*args=None*)

clowdr.driver.**makeparser**()

Command-line API wrapper for Clowdr as a CLI, not Python API. For information about the command-line wrapper and arguments it accepts, please try running "clowdr –help".

args: list List of all command-line arguments being passed.

int The exit-code returned by the driver.

clowdr.driver.**runtask**(*tasklist*, ***kwargs*)

clowdr.driver.**share**(*provdir*, *prepare=False*, *host='0.0.0.0'*, *port=8050*, *verbose=False*, *debug=False*, ***kwargs*)

Launches a simple web server which showcases all runs at the clowdrloc.

provdir [str] Path with Clowdr metdata files (returned from "local" and "deploy")

****kwargs** [dict] Arbitrary keyword arguments (i.e. {'verbose': True})

None

5.1.4 clowdr.server module

5.1.5 clowdr.task module

class clowdr.task.**TaskHandler**(*taskfile*, ***kwargs*)

Bases: object

execWrapper(*sender*)

manageTask(*taskfile*, *provdir=None*, *verbose=False*, ***kwargs*)

monitor(*target*, ***kwargs*)

provLaunch(*options*, ***kwargs*)

5.1.6 clowdr.utils module

clowdr.utils.**backoff**(*function*, *posargs*, *optargs*, *backoff_time=36000*, *verbose=False*, ***kwargs*)

clowdr.utils.**get**(*remote*, *local*, *verbose=False*, ***kwargs*)

clowdr.utils.**getContainer**(*savedir*, *container*, *simg=None*, *verbose=False*, ***kwargs*)

clowdr.utils.**post**(*local*, *remote*, *verbose=False*, ***kwargs*)

clowdr.utils.**randstring**(*k*)

clowdr.utils.**remove**(*local*)

clowdr.utils.**splitS3Path**(*path*)

clowdr.utils.**truepath**(*path*)

CHAPTER 6

Indices and tables

- genindex
- search

Python Module Index

C

Index